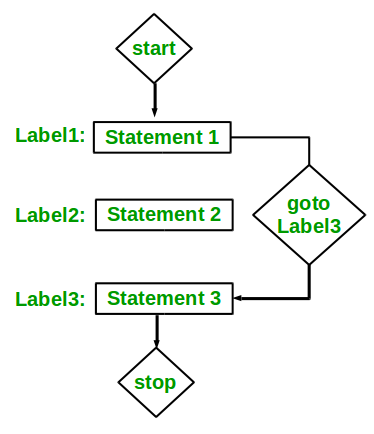
**goto Statement in C**

The **C goto statement** is a jump statement which is sometimes also referred to as an **unconditional jump** statement. The goto statement can be used to jump from anywhere to anywhere within a function.

In the above syntax, the first line tells the compiler to go to or jump to the statement marked as a label. Here, the label is a user-defined identifier that indicates the target statement. The statement immediately followed after ‘label:’ is the destination statement. The ‘label:’ can also appear before the ‘goto label;’ statement in the above syntax.



*Flowchart of goto Statement in C*

Below are some examples of how to use a goto statement.

**Examples:**

**Type 1**: In this case, we will see a situation similar to as shown in Syntax1 above. Suppose we need to write a program where we need to check if a number is even or not and print accordingly using the goto statement. The below program explains how to do this:

|  |
| --- |
| // C program to check if a number is  // even or not using goto statement  #include <stdio.h>    // function to check even or not  **void** checkEvenOrNot(**int** num)  {  **if** (num % 2 == 0)          // jump to even  **goto** even;  **else**          // jump to odd  **goto** odd;    even:  **printf**("%d is even", num);      // return if even  **return**;  odd:  **printf**("%d is odd", num);  }    **int** main()  {  **int** num = 26;      checkEvenOrNot(num);  **return** 0;  } |

**Output:**

26 is even

**Type 2:** In this case, we will see a situation similar to as shown in Syntax2 above. Suppose we need to write a program that prints numbers from 1 to 10 using the goto statement. The below program explains how to do this.

|  |
| --- |
| // C program to print numbers  // from 1 to 10 using goto statement  #include <stdio.h>    // function to print numbers from 1 to 10  **void** printNumbers()  {  **int** n = 1;  label:  **printf**("%d ", n);      n++;  **if** (n <= 10)  **goto** label;  }    // Driver program to test above function  **int** main()  {      printNumbers();  **return** 0;  } |

**Output:**

1 2 3 4 5 6 7 8 9 10

**Disadvantages of Using goto Statement**

* The use of the goto statement is highly discouraged as it makes the program logic very complex.
* The use of goto makes tracing the flow of the program very difficult.
* The use of goto makes the task of analyzing and verifying the correctness of programs (particularly those involving loops) very difficult.
* The use of goto can be simply avoided by using [break](https://www.geeksforgeeks.org/break-statement-cc/) and [continue](https://www.geeksforgeeks.org/continue-statement-cpp/) statements.